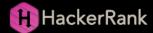
# **Array Manipulation**



Starting with a 1-indexed array of zeros and a list of operations, for each operation add a value to each of the array element between two given indices, inclusive. Once all operations have been performed, return the maximum value in your array.

For example, the length of your array of zeros n=10. Your list of queries is as follows:

```
a b k
153
487
691
```

Add the values of k between the indices a and b inclusive:

```
index-> 1 2 3 4 5 6 7 8 9 10
[0,0,0, 0, 0,0,0,0,0, 0]
[3,3,3, 3, 3,0,0,0,0, 0]
[3,3,3,10,10,7,7,7,0, 0]
[3,3,3,10,10,8,8,8,1, 0]
```

The largest value is 10 after all operations are performed.

#### **Function Description**

Complete the function *arrayManipulation* in the editor below. It must return an integer, the maximum value in the resulting array.

arrayManipulation has the following parameters:

- *n* the number of elements in your array
- queries a two dimensional array of queries where each queries[i] contains three integers, a, b, and k.

#### **Input Format**

The first line contains two space-separated integers n and m, the size of the array and the number of operations. Each of the next m lines contains three space-separated integers a, b and k, the left index, right index and summand.

#### **Constraints**

- $3 < n < 10^7$
- $1 \le m \le 2 * 10^5$
- $1 \le a \le b \le n$
- $0 < k < 10^9$

#### **Output Format**

Return the integer maximum value in the finished array.

#### Sample Input

```
5 3
1 2 100
2 5 100
3 4 100
```

### **Sample Output**

## **Explanation**

After the first update list will be 100 100 0 0.

After the second update list will be 100 200 100 100 100.

After the third update list will be 100 200 200 200 100.

The required answer will be 200.